Deleted: DRAFT

THE 2004 TOPIC DETECTION AND TRACKING (TDT2004) TASK DEFINITION AND EVALUATION PLAN

1. INTRODUCTION

The purpose of the TDT project is to advance the state of the art in Topic Detection and Tracking. The task domain is multilingual human language. This domain is to be explored and technology is to be developed in the context of an evaluation-guided R&D paradigm in which key technical challenges are defined and supported by common informative evaluations. This document defines the tasks, the performance metrics, and the evaluation procedures to be used to direct the research and to evaluate research progress and technical capabilities.

TDT addresses multiple sources of information, including both text and speech. These sources are namely newswires, radio and television news broadcast programs, and WWW sources. The source languages are English, Mandarin and Arabic. The information flowing from each source is modeled as a sequence of stories. These stories provide information on many topics. The general technical challenge is to identify and to follow the topics being discussed in these stories.

2. TOPIC DEFINITION

In the initial TDT study, conducted during 1996 and 1997, the notion of a topic was limited to be an "event", meaning something that happens at some specific time and place. For example, the eruption of Mount Pinatubo on June 15th, 1991 is considered to be an event, whereas volcanic eruption in general is not. Events might be unexpected, such as an airplane crash, or expected, such as a political election.

In the second TDT project, TDT2, the definition of a topic was broadened to include, in addition to the triggering event, other events and activities that are directly related to it. This definition was retained for the third project, TDT3, TDT4 and TDT5:

> ***A topic is defined to be a seminal event or activity, along with all directly related events and activities***.

A story will be considered to be "on topic" whenever it discusses events and activities that are *directly* connected to that topic's seminal event. So, for example, a story on the search for survivors of an airplane crash, or on the funeral of the crash victims, will be considered to be a story on the crash event. Obviously there must be limits to this inclusiveness. (For example, stories on FAA repair directives that derive from a crash investigation would not be considered to be stories on the crash event.) Topic boundaries are subject to interpretation, so the Linguistic Data Consortium (LDC) has created guidelines to improve agreement and consistency of topic labeling.[1]

[1] Determining the limits of TDT topics is often extremely difficult and arbitrary. The question is where to draw the line on including (or excluding) "related" events. The LDC have facilitated this task considerably by identifying certain general types of topics and creating specific boundary determination rules for each of those topic types. These rules are contained in a web-accessible document that LDC uses to instruct and guide annotation. Annotator guidelines for the current TDT corpus can be found at http://ldc.upenn.edu/Projects/TDT5/.

3. TDT CORPORA

LDC[2] is providing five corpora to support TDT research: the TDT Pilot corpus and the TDT2, TDT3, TDT4 and TDT5 corpora. These corpora are collections of news from a number of sources and languages. All of these corpora contain news from print sources, and all but TDT5 also include broadcast news.

The TDT2 corpus spans the first six months of 1998 and consists of English and Mandarin data. It is annotated for 200 topics in English, 20 of which have also been annotated in Mandarin.

The TDT3 corpus spans October-December 1998 and consists of English, Mandarin and Arabic data. There are 120 annotated topics for the English and Mandarin data. A subset of these topics are also annotated for the Arabic data.

The TDT4 corpus spans October 2000-January 2001 and consists of English, Mandarin and Arabic data. There are 80 topics annotated for the corpus in all three languages.

The TDT5 corpus spans April-September 2003 and consists of English, Mandarin, and Arabic data. There are 250 topics annotated for the corpus. Approximately 25% of these are annotated for all three languages; the remainder are annotated for only one language, evenly distributed across the three languages. Topics annotated in only one language are chosen to be "local" topics unlikely to appear in sources in the other languages.

Each story in the TDT2 and TDT3 corpora is tagged according to whether it discusses each of the defined topics. These story-topic tags (Tag[story,topic]) assume a value of YES if the story discusses the target topic, BRIEF if that discussion comprises less than 10% of the story, or otherwise NO (the default tag) if the story does not discuss the topic.

Each story in the TDT4 and TDT5 corpora is tagged YES if the story discusses the target topic, and NO otherwise. A story tagged YES will contain some information about the topic, not merely a reference to it, but the relevant material may be either brief or lengthy.

The TDT5 topic annotation differs from previous corpora in that topics are not necessarily fully annotated. In previous corpora, annotators followed a protocol designed to locate *all* stories on each topic. But for TDT5, annotators have a fixed time allocation for each topic, and follow a protocol designed to locate as many stories as possible within the time allowed. As a result, there are likely to be some relevant stories that were not found, particularly for topics encompassing a large number of stories. For each topic, the annotator indicates whether the annotation seems to be complete or incomplete.

For the corpora containing broadcast news material, LDC provides three different representations of the data:

1. The audio sampled data signal.
2. A manual transcription of the audio signal.
3. A transcription produced automatically by an automatic speech recognition (ASR) system.

[2] The Linguistic Data Consortium Phone: 215/898-0464
Email: ldc@ldc.upenn.edu Fax: 215/573-2175
URL: http://www.ldc.upenn.edu/

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

The transcriptions include non-news material in addition to news stories. (Non-news stories include commercials and list-type reports such as sports scores and financial data.) Accordingly, broadcast stories are labeled either as NEWS or MISCELLANEOUS or, in case there exists no transcription for a story, as UNTRANSCRIBED.

For Arabic and Mandarin sources, there are two different text representations:

1. The original language character source stream, which is either source text (for newswire), or a manual or ASR transcription of speech (for broadcast news data).[3]
2. An English translation produced automatically from the original language source stream.

For complete details on the TDT2, TDT3 and TDT4 corpora, refer to URL http://www.ldc.upenn.edu/TDT/.

Further information on the TDT5 corpus is available at URL http://www.ldc.upenn.edu/Projects/TDT5.

3.1 CORPUS RESOURCES FOR TDT 2004

The TDT-Pilot, TDT2, TDT3 and TDT4 corpora are all designated as training resources for the 2004 TDT Evaluation. Systems may make use of these corpora in any way.

The 2004 TDT Evaluation will use the TDT5 corpus as the test corpus. As such, no participants may train on the TDT5 corpus. The evaluation corpus will be shipped to the new participating research sites as specified by the TDT schedule.[4]

Participants may supplement the TDT training corpora with any other data; however, all additional data must predate the evaluation corpus which begins April 1, 2003.

4. THE TASKS

There are four TDT tasks defined for the 2003 evaluation: the tracking of known topics, the detection of unknown topics, the detection of initial stories on unknown topics, and the detection of pairs of stories on the same topic (links). Of these four tasks, the topic tracking task and the link detection task are considered to be "primary." All sites that choose to participate in the evaluation will be required to perform at least one of these primary tasks, and one or both of these tasks should be the primary focus of sites' TDT research. That is because these tasks represent core technology that is broadly applicable to many different TDT applications.

Previous TDT evaluations included a story segmentation task. This task applied only to broadcast news. Since TDT5 does not include broadcast news, there is no story segmentation task in the 2004 TDT Evaluation.

4.1 THE TOPIC TRACKING TASK (PRIMARY)

The TDT topic tracking task is defined to be the task of associating incoming stories with topics that are known to the system. A topic is "known" by its association with stories that discuss it. Thus each target topic is defined by one or more stories that are "on" (i.e., that discuss) the topic. To support this task, a small set of on-topic training stories is identified for each topic to be tracked. The system may train on the target topic by using all of the stories in the corpus, up through the most recent training story. The tracking task is then to classify correctly all subsequent stories as to whether or not they discuss the target topic.

4.2 THE SUPERVISED ADAPTIVE TRACKING TASK

An optional variant of the topic tracking task is supervised adaptive tracking. This task is identical to the topic tracking task except that, for each story judged to be on-topic, the relevance judgment for that story is then made available, allowing supervised adaptation during tracking. This task is very similar to the TREC adaptive filtering track.

4.3 THE HIERARCHICAL TOPIC DETECTION TASK

Previous TDT evaluations have included a Topic Detection Task. For the 2004 TDT Evaluation, this task has been replaced by a Hierarchical Topic Detection Task. This task is described fully in a supplement to the 2004 TDT Evaluation Plan, incorporated in this document as Appendix A.

Because this is the first attempt at evaluating Hierarchical Topic Detection, this evaluation will be regarded as an experimental, "dry run" evaluation.

4.4 THE NEW EVENT DETECTION TASK

The TDT new event detection task is defined to be the task of detecting, in a chronologically ordered stream of stories from multiple sources (and in multiple languages), the first story that discusses an event. This task may be viewed as being essentially the same as the (non-hierarchical) topic detection task. The principal difference is in what the detection system outputs.

4.5 THE LINK DETECTION TASK (PRIMARY)

The TDT link detection task is defined to be the task of determining whether two stories discuss the same topic. Thus, the system must embody an understanding of what a topic is, and this understanding must be *independent of topic specifics*. The link detection task, however, does not deal with topics explicitly. Thus links are not constrained to segregate stories into a set of orthogonal topics, and there is no presumption that each story discusses one and only one topic.

5. THE EVALUATION

In order to inform TDT research, to guide TDT technology development, and to assess TDT application potential, TDT task performance will be evaluated according to a set of rules for each of the four TDT tasks.

Evaluation methodology, parameters and procedures for 2004 are similar to those for 2003. There are some necessary differences because of the changes in annotation protocol and the elimination of broadcast news in TDT5. But in other respects, the evaluation remains unchanged for the three continuing tasks: Topic Tracking, First-Story Detection, and Link Detection. The evaluation procedures for the new task, Hierarchical Topic Detection, build on previous procedures.

[3] The Mandarin ASR transcription includes whitespace delimitation of words. Original source text and manual transcriptions do not.

[4] http://www.nist.gov/speech/tests/tdt/tdt2004/sched.htm

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

All of the TDT tasks are cast as detection tasks. Detection performance is characterized in terms of the probability of miss and false alarm errors (P_{Miss} and P_{FA}). These error probabilities are then combined into a single detection cost, C_{Det}, by assigning costs to miss and false alarm errors:

$$C_{Det} = C_{Miss} \cdot P_{Miss} \cdot P_{target} + C_{FA} \cdot P_{FA} \cdot P_{non-target}$$

where

- C_{Miss} and C_{FA} are the costs of a Miss and a False Alarm, respectively,
- P_{Miss} and P_{FA} are the conditional probabilities of a Miss and a False Alarm, respectively, and
- P_{target} and $P_{non-target}$ are the *a priori* target probabilities ($P_{non-target} = 1 - P_{target}$).

C_{Det} is the bottom-line representation of TDT task performance that is used to judge TDT systems. This cost measure is often a reasonable measure of application value, and consideration of the application can provide appropriate values for the relative costs of misses and false alarms and the target probability. Because these values vary with the application, C_{Det} will be normalized so that $(C_{Det})_{Norm}$ can be no less than one without extracting information from the source data. This is done as follows:

$$(C_{Det})_{Norm} = C_{Det} \,/\, \min(C_{Miss} \cdot P_{target}, C_{FA} \cdot P_{non-target})$$

Thus the absolute value of $(C_{Det})_{Norm}$ is a direct measure of the value (i.e., relative cost) of the TDT system.

Hierarchical Topic Detection uses alternative cost measures based on C_{Det}, as described in Appendix A.

There are two reasonable methods of estimating detection error probabilities, called story-weighted and topic-weighted. The story-weighted method assigns equal weight to each decision for each story and accumulates errors over all topics. The topic-weighted method accumulates errors separately for each topic and then averages the error probabilities over topics, with equal weight assigned to each topic. **The topic-weighted method will be used exclusively in TDT2004**, because it provides better estimates of performance.[5]

Input data to the TDT systems comprises the various news sources. This source stream is presented to the processing systems in chronological order, and the various sources are presented together, interwoven so as to preserve chronological order. Source file sequencing will be controlled by means of a list of chronologically ordered source file names. Each source file will contain an uninterrupted sample of source data. It is assumed that there is no temporal overlap between different source files.[6]

[5] A major source of variance in error probability estimates is the topic. Therefore, because of the high variability in the number of stories per topic, it is important to reduce the contribution of topic variance by equalizing the contribution of different topics.

[6] It is certain that data in different source files will overlap occasionally. The assumption of no overlap is made, however, because of the great simplification provided. The loss of strict temporal order is judged minor and insignificant, because the time duration represented by each source file is a small fraction of a single day.

For all four TDT tasks, the system may use knowledge of the source of the data and knowledge of the time of the stories.

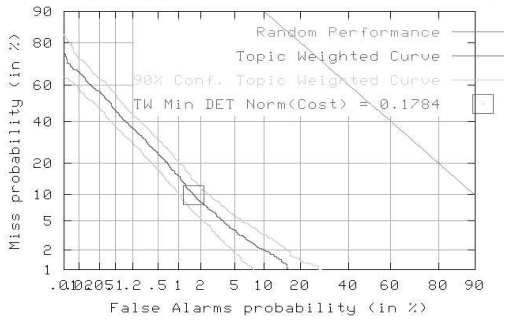
5.1 Detection Error Tradeoff Curves

Detection Error Tradeoff (DET) curves are visualizations of the tradeoff between of missed detection (P_{Miss}) rate and the false alarm (P_{Fa}) rate. The curves are constructed by sweeping a threshold through the system's space of decision scores. At each point in the score space, P_{Miss} and P_{Fa} are estimated and plotted as a connected line.

This method generates a story-weighted DET curve. Story-weighted DET curves suffer from the same vulnerabilities as story-weighted measures discussed earlier, so TDT uses a topic-weighted DET curve to match the topic-weighted $(C_{Det})_{Norm}$. Topic weighted DET curves are made as follows: sort the stories in order of decision scores separately for each topic. Again, step through the score space, but rather than calculate global P_{Miss} and P_{Fa}, compute the average of P_{Miss} and P_{Fa} across topics. Since means are estimated, variances can also be computed which allows computation of confidence region.

Figure 1 is a DET curve from the 2003 tracking evaluation. The Y-axis is the probability of missed detection and the X-axis is the probability of false alarms. Since missed detections and false alarms are types of errors, improvements in performance will be shown by lines moving closer to the lower left hand corner. Note that the normal deviant scale (expressed as percentages) is used on both axes. The normal deviant scale has advantages over linear scales. It expands the "high performance" region, and resulting straight lines indicate normality of the underlying error distributions of P_{Miss} and P_{Fa}.

Figure 1. Example DET Curve from 2003 Tracking Evaluation



5.2 Topic Tracking Evaluation

Tracking algorithms will be evaluated in terms of their ability to detect which stories are on-topic and which are not. Topics are to be tracked individually, and each topic is to be treated separately and independently. In training the system for a particular target topic, allowable information includes the training set and topic tags *for that target topic only*. During the evaluation of each target topic, no information is given on any other topic. Evaluation will be over the whole extent of the evaluation corpus.

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

For each topic tracked, the primary system output will be a DECISION (YES/NO) for each story processed, indicating whether the story is judged to be on-topic or not. In addition to this primary output, the system must also produce a SCORE that indicates the confidence that the story is on-topic for the target topic. This SCORE may then be used to explore the trade-off between miss and false alarm errors. It may also be used to guide post-test annotation adjudication.

A primary task parameter is N_t, the number of stories used to define the target topic. The maximum value of N_t, $(N_t)_{Max}$, will be 4. All of these on-topic training stories are tagged YES for the target topic. The evaluation corpus will be divided chronologically into a different training set and test set for each target topic. The training set will comprise the first part of the corpus, up through the last training story tagged YES for the target topic. In addition, the training data will include story identification of the N_t stories that serve to define the target topic. The test set will comprise the remainder of the corpus that follows. Note that the target topic is defined using the *last* N_t on-topic training stories in the training set for the target topic.

Another important issue is which language(s) are to be used for topic training. Topic training may be performed on English, Mandarin or Arabic sources, but not two at once. Note that for TDT5, 25% of the topics are annotated in all three languages, and 25% are annotated in each of the other three languages. Therefore, 50% of the topics are available for training in any one language.

Evaluation will be supported for three values of N_t. The various possible combinations are shown in Table 1.

Table 1. **Training conditions for topic tracking evaluation**

	English	Mandarin	Arabic
N_t – the # of on-topic stories	1	1	1
	2	2	2
	4	4	4
	Variable, 1-4	Variable, 1-4	Variable, 1-4

Note that there is also a variable N_t condition in Table 1. This condition simulates application situations in which the number of training stories is not controlled. It also represents a conceptually more difficult task, because optimum decisions require that the decision score be normalized across different values of N_t.

Note that there is no knowledge of which cross-language stories in the training data are on-topic, because training is monolingual. This applies to both multilingual topics (those annotated in all three languages) and monolingual topics (those annotated in only one language). For monolingual topics, all cross-language stories are off-topic, but sites will not be informed which topics are monolingual.

Sites are permitted to use the cross-language portion of the training data for training purposes, albeit *without* any on-topic story tags being given for cross-language data. Note also that stories not tagged as on-topic are not guaranteed to be off-topic, even in the training data for the given training language.

Independent of the language training condition, the *test* set for each topic will include *all* sources for *all* languages. Evaluation of topic tracking performance will be conditioned on source language. This conditional analysis of tracking output will provide performance results as a function of language and will measure performance separately for cross-language and same-language training/test conditions. Also, performance will be reported separately for multilingual and monolingual topics.

The topic tracking system must conform to the following rules:

- The topic tracking system must process the input source data in chronological order.
- Tracking output must be made as each story is processed.
- Unsupervised adaptation is allowed, using information obtained as the input data are processed.
- No look-ahead is allowed.

Each research site is encouraged to study as many of the above conditions as may be productively done. However, all sites that perform the tracking evaluation are required to perform the required condition. This condition is shown in Table 2.

Table 2. **Required Topic Tracking Condition**

Parameter	Value
Topic training language	English
On-topic training stories	1

Topic tracking performance will be measured using the detection cost formula to combine P_{Miss} and P_{FA}. The evaluation cost parameters to be used for the TDT2004 evaluation are given in Table 3. Choice of P_{target} was based on an analysis of the stories in the TDT2 training corpus.

Table 3. **Topic tracking evaluation cost parameters**

Parameter	Value
P_{target}	0.02
C_{Miss}	1.0
C_{FA}	0.1

5.3 SUPERVISED ADAPTIVE TRACKING EVALUATION

The supervised adaptive tracking evaluation is identical to the topic tracking evaluation described in 5.2, except that relevance feedback is available for stories judged to be on-topic.

Sites participating in this evaluation will receive a reference file containing annotator judgment information for each story for each topic. More specifically, the reference file will contain an entry for each story judged by a human annotator, indicating whether the story was labeled on-topic or off-topic. Stories that have not been judged will not appear in the reference file. For scoring purposes, unjudged stories are treated as off-topic. For supervised adaptation, it is up to sites to decide how to use the feedback information, and whether to treat unjudged stories like off-topic stories or to distinguish these classes.

The reference file is to be used as follows:

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

- For each test story for each topic, if the system decision on the story is YES (i.e., the story is deemed on-topic), then the system may immediately access the reference file to determine whether the story was labeled on-topic, off-topic, or unjudged. This relevance feedback information may then be used for adaptation.
- If the system decision on the story is NO (not on-topic), then the system may not access the reference file for that story and may not use relevance feedback on that story for adaptation.

This evaluation scenario corresponds to an application scenario in which the system routes on-topic stories to a user, who then provides relevance judgments that are (immediately) available as feedback to the system.

In all other respects, the conditions for supervised adaptive tracking are identical to those for topic tracking, as described in 5.2. Conditions for initial training and treatment of languages are subject to the same optional and required conditions shown in Tables 1 and 2.

Performance on supervised adaptive tracking will be measured in two ways. The output of each system will be scored using the TDT topic tracking metrics and parameters, exactly as described in section 5.2. This will provide data for comparison to the unsupervised tracking task.

In addition, performance on supervised adaptive tracking will be measured using the ***linear utility measure*** as defined in TREC 2002[7] (the last TREC evaluation that included an adaptive filtering track). The basic linear utility measure is:

$$U = W_{Rel} * R - NR$$

where R = number of relevant documents retrieved, NR = number of non-relevant documents retrieved, and W_{Rel} is a constant that determines the relative weighting of relevant vs. non-relevant documents in determining the utility score. For consistency with the detection cost metric parameter values, which effectively weight relevance as 10 times more important than non-relevance (C_{Miss} vs. C_{FA}), a value of 10 will be used for W_{Rel}.

As Robertson and Soboroff explain, there are two potential problems with the basic utility measure. First, if average utility across topics is computed using this measure, the result will be dominated by topics for which many documents are retrieved. This is analogous to story-weighted computation of detection cost. To make the measure analogous to topic-weighted computation, in which each topic contributes equally to the overall score, the utility measure is normalized by the maximum possibly utility for the topic:

$$U_{Norm} = U / U_{Max}$$

where $U_{Max} = W_{Rel} * \textit{total number of relevant docs}$

A second potential problem is that the number of non-relevant documents retrieved could be huge, relative to the number of relevant documents. This means that the overall score could be dominated by one or a small number of topics with exceptionally poor scores. To avoid this, a floor is applied by defining a minimum utility value. If the score for a document falls below this minimum, the minimum value is used instead. This corresponds to an application scenario in which users stop looking at documents when the number of non-relevant documents exceeds some threshold. Following TREC 2002, the value of U_{Min} will be set at –0.5.

Using both normalization and the minimum score floor, the utility score is then scaled to fall between 0 and 1, as follows:

$$U_{Scale} = [\ max(U_{Norm}, U_{Min}) - U_{Min}\ ] / [\ 1 - U_{Min}\ ]$$

This scaled, normalized linear utility measure will be used in the 2004 evaluation. The parameter values for the 2004 evaluation are summarized in Table 4.

Table 4: **Supervised Adaptive Tracking utility measure parameter values**

Parameter	Value
W_{Rel}	10
U_{Min}	-0.5

One purpose of the 2004 evaluation is to explore the relationship between the two types of measures, i.e., the detection cost and DET curve measure vs. the utility measure. Sites may optimize for either measure, or may submit two different system outputs, one optimized for each measure. For each submission, sites are requested to inform NIST of the measure for which it was optimized.

5.4 HIERARCHICAL TOPIC DETECTION EVALUATION

A detailed description of the Hierarchical Topic Detection Task is contained in a supplement to the 2004 TDT Evaluation Plan, included as Appendix A.

5.5 NEW EVENT DETECTION EVALUATION

The new event detection task is logically the same as the non-hierarchical topic detection task.[8] The evaluation, however, focuses on the specific aspect of detection associated with novel information. To aid in new event detection research, the TDT3 corpus has been augmented with new event annotation for an additional 120 topics.

The primary new event detection system output will be a DECISION (YES/NO) for each story processed, indicating whether the story is judged to be the first story of a topic or not. In addition to this primary output, the system must also produce a SCORE that indicates the confidence that the story is a first story. This SCORE may then be used to explore the trade-off between miss and false alarm errors. It may also be used to guide post-test annotation adjudication.

Evaluation will be over the entire portion of the evaluation corpus for which the source language is English. Evaluation over Arabic and Mandarin data is not included in this task.

The new event detection system must process the input source data in chronological order. However, the new event detection

[7] Stephen Robertson and Ian Soboroff, "The TREC 2002 Filtering Track Report." This document is available for download at trec.nist.gov/pubs/trec11/papers/OVER.FILTERING.pdf

[8] Knowing when a story is the first story on a topic implies knowing when a story is *not* a first story.

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

system may defer its decision until a limited amount of subsequent source data is processed. This deferral period, N_f, is a primary task parameter and is the number of source files, including the source file being processed, for which processing may be completed before committing to and outputting a new event detection decision. (The greater the deferral, presumably the better will be decisions.) The deferral parameter values to be used in TDT2004 are shown in Table 4.

Table 4. **Maximum decision deferral periods for new event detection evaluation**

# of source files, including the one being processed
1
10 *(required)*
100

Each research site is encouraged to study as many of the above conditions as may be productively done. However, all sites that perform the new event detection evaluation are required to perform at least one evaluation under common shared conditions for the task. These conditions are given in Table 5.

Table 5. **Required New Event Detection Conditions**

Parameter	Values
Source Language	English only
Maximum decision deferral period	10 source files

New event detection performance will be measured using the detection cost formula to combine P_{Miss} and P_{FA}. The evaluation cost parameters to be used for the TDT2004 evaluation are given in Table 6. Choice of P_{target} was based on an analysis of the stories in the TDT2 training corpus.

Table 6. **New event detection evaluation cost parameters**

Parameter	Value
P_{target}	0.02
C_{Miss}	1.0
C_{FA}	0.1

5.6 LINK DETECTION EVALUATION

Link detection algorithms will be evaluated in terms of their ability to determine whether specified pairs of stories discuss the same topic (i.e., are "linked"). The pairs of stories may be in the same language or in different languages; in either case, the task is the same.

The primary link detection system output will be a DECISION (YES/NO) for each pair of stories processed, indicating whether the stories are judged to be on the same topic or not. In addition to this primary output, the system must also produce a SCORE that indicates the system's confidence that the story-pair is linked. This SCORE may then be used to explore the trade-off between miss and false alarm errors. It may also be used to guide post-test annotation adjudication.

The link detection task will use English, Mandarin and Arabic source data. Evaluation of link detection will be conditioned on source language. This conditional analysis of link detection performance will tabulate performance separately for cross-language and same-language story pairs.

The link detection system must process the input source data in chronological order. However, the link detection system may defer its identification of story links until a limited amount of subsequent source data is processed. This deferral period, N_f, is a primary task parameter and is the number of source files, including the source file being processed, for which processing may be completed before making story-story link decisions. (The greater the deferral, presumably the better will be the link decisions.) The deferral parameter values to be used in TDT2004 are shown in Table 4.

Each research site is encouraged to study as many of the above conditions as may be productively done. However, all sites that perform the link detection evaluation are required to perform at least one evaluation under common shared conditions for the link detection task. These conditions are shown in Table 7.

Table 7. **Required Link Detection Conditions**

Parameter	Value
Source Language	English, Mandarin and Arabic
Maximum decision deferral period	10 source files

While link detection systems are expected to be capable of making link decisions for all pairs of stories, evaluation of all such decisions is neither practical nor necessary. Therefore evaluation will be limited to a subset of story pairs sufficient to provide reliable estimates of P_{Miss} and P_{FA}. This will keep system output files to a manageable level. System output must be limited to the specified story pairs. The story pairs for which output is required will be listed in chronological order, ordered primarily according to the second (i.e., the most recent or newest) story of the pair and secondarily according to the first (i.e., the oldest) story of the pair. This will facilitate coordination of system output with chronological processing and deferral requirements. The maximum deferral period is with respect to the source file containing the most recent of the pair of stories.

Formatted: Bullets and Numbering

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

Link detection performance will be measured using the detection cost formula to combine P_{Miss} and P_{FA}. For the link detection task, P_{target} is the probability that a pair of stories chosen at random discuss the same topic. The evaluation cost parameters to be used for the TDT2004 evaluation are given in Table 9.

Table 9. **Link detection evaluation cost parameters**

Parameter	Value
P_{target}	0.02
C_{Miss}	1.0
C_{FA}	0.1

6. ADJUDICATION

After NIST determines the initial results for all systems on these TDT evaluation tasks, LDC will review and adjudicate selected cases. Annotation adjudication is likely to have more impact this year than in previous years because of the time-constrained annotation protocol for TDT5. With this protocol, some topics are not completely annotated.

Candidates for adjudication will be selected based on system output. Specifically, false alarm errors will be re-examined to determine whether the apparent false alarm is actually an annotation miss. Since there will be far more false alarm errors than LDC can re-examine, candidate errors will be selected based on the following: (a) errors made by all or most systems; (b) errors involving stories marked NO by default, not by examination; (c) errors involving topics marked INCOMPLETE; (d) errors involving cross-language stories for monolingual topics, and (e) errors that ranked high in terms of system output confidence scores.

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

APPENDIX A: THE 2004 HIERARCHICAL TOPIC DETECTION TASK DEFINITION AND EVALUATION PLAN

1. INTRODUCTION

This document is a supplement to the 2004 Topic Detection and Tracking Task Definition and Evaluation Plan.[9] This document describes a plan for an initial, trial evaluation of hierarchical topic detection (HTD). It addresses only those aspects of the evaluation that are specific to HTD. For all aspects of the evaluation not discussed here, the HTD evaluation will follow the same procedures as other TDT2004 tasks.

2. BACKGROUND

The hierarchical approach to the topic detection task is intended to overcome two problematic assumptions in conventional topic detection: the assumption that all topics are at the same level of granularity, and the assumption that each story pertains to at most one topic. In fact, topics are often at different levels of granularity, e.g. the Asian Economic Crisis vs. the G-7 World Finance Meeting, yet systems have been required to select a single operating point to maximize performance on all topics. Some stories pertain to more than one topic, but systems have assigned stories to non-overlapping clusters, and stories judged to pertain to multiple topics have been discarded in evaluation.

Hierarchical topic detection will allow stories to belong to multiple clusters and will allow clusters to be defined at different levels of granularity. This should allow more meaningful evaluation and enable more progress in automatic document clustering by topic.

3. THE TASK

The HTD task is to automatically cluster a collection of stories by topic, with the resulting set of clusters having the following properties:

- Every story is assigned to one or more clusters.
- Clusters may be subsets of other clusters or may overlap with other clusters.
- The relationships among clusters must be characterizable as a directed acyclic graph (DAG) with a single root node (described further below).

3.1 CONDITIONS

For this initial trial evaluation, the task involves a simplification in its treatment of time. Time synchrony of clustering decisions is not required, and there is no maximum deferral period. The task is treated as retrospective search.

The language conditions are based on the customary topic detection task requirements, with minor modifications. The multilingual condition (English, Mandarin and Arabic) will be required of all systems, as usual for topic detection. In addition, the English only task will be required of all systems, in order to ensure that multiple systems are evaluated under a monolingual condition. Since this is a trial evaluation, it is important to obtain sufficient participation to exercise the evaluation process and metrics adequately. The remaining tasks, Mandarin only and Arabic only, will continue to be optional. This is summarized in Table A1.

Table A1. **Language conditions for HTD evaluation**

Language Conditions
English only *(required)*
Mandarin only
Arabic only
English, Mandarin and Arabic together *(required)*

For the multilingual task, scores conditioned on the individual languages will not be computed for this initial trial evaluation.

3.2 DAG SEMANTICS

Each HTD system will construct a DAG over the designated collection of topics. The root vertex of the DAG represents the entire collection.[10] Children of the root represent subsets of stories (which may be overlapping). At each successive layer of the DAG, vertices represent subsets of their parent clusters. Again, each subset may overlap with other subsets. Thus, the layers of the DAG represent increasing granularity, with the root vertex being most general (i.e., the entire collection), and the leaf vertices being most specific.

Alternatively, one can view the DAG from the leaves to the root. Each leaf cluster represents a maximally specific topic. Each parent of a leaf represents a somewhat more general topic that subsumes the leaf topic. As one ascends the DAG, each parent cluster is increasingly general, terminating in the maximally general root vertex that includes the entire collection.

Because the structure is a DAG, not a tree, a cluster can be a subset of more than one more general cluster.

3.3 DAG TOPOLOGY

HTD systems will produce DAGs with this topology:

- Vertices are identified by a unique string.
- Each vertex contains a list of stories and/or pointers to one or more other vertices.

[9] http://www.nist.gov/speech/tests/tdt/tdt2004/evalplan.htm

[10] The root vertex does not necessarily represent a topically coherent cluster, since it contains the entire collection. However, it is useful pragmatically because it allows for navigation from any story or cluster to any other story or cluster in the collection.

Deleted: . HTD Task Definition and Evaluation Plan . page

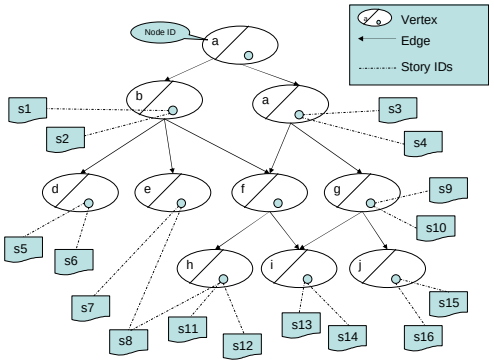
Deleted: of

Deleted: . version 1.0,

- All stories in the list for a given vertex are automatically included in the lists for all precedessors of that vertex.
- Stories can be in any vertex story list (except that redundant inclusion in predecessor vertices' lists is disallowed).
- Pointers to other vertices must not create cycles.
- There is a single vertex, designated the root vertex, that is the predecessor of all other vertices.

Figure A1 illustrates a system-generated DAG.

Figure A1. Example DAG of topic clusters



As this example illustrates, stories will be identified by the story identifier, not by segmentation indices.

The DAG of clusters will be represented in XML format. See the Appendix on Implementation Details for the exact format specification.

4. THE EVALUATION

4.1 EVALUATION METRIC

There are a variety of possible metrics for hierarchical topic detection. Several of these are described and analyzed in Allan et al.[11] One of these, the Minimal Cost metric, has been generalized and will be used as the primary metric for the trial HDT evaluation. NIST will also examine the behavior of other metrics on the trial data to the extent time permits.

For the Minimal Cost metric, the score for each topic is a linear combination of the normalized detection cost and normalized travel cost. These scores per topic are then averaged to compute an overall performance score.

The Minimal Cost metric has the following desirable characteristics:

- The power set is effectively eliminated as a potential solution.
- It is conceptually appealing in that it drives systems to find a balance between two opposing modes: creating huge numbers of clusters (which increases the probability of having a cluster that exactly matches a topic) vs. creating only a few clusters (which reduces the cost of finding the best-matching cluster).
- It demonstrated good behavior in initial experiments, as reported in Allan et al. Specifically, the increase in cost as the optimal cluster match degrades is approximately linear. Also, with appropriate choices of constants, the metric favors a good hierarchical structure over a flat one.
- The algorithm is computationally tractable. In the worst case, it requires O(nm), where n is the number of topics and m is the number of nodes. Better performance is obtained by pruning the search after the cost of detection for a child vertex is larger than the cost of detection for its parent.

The following formulas define the Minimum Cost metric for HTD:

MinimumCost(*topic*) =

WDET*(Cdet(topic,*bestVtx*))$_{Norm}$ +

(1–**WDET)***(Ctravel(topic,*bestVtx*))$_{Norm}$

WDET is an evaluation constant used to set the relative weights assigned to each type of cost. For the evaluation, more weight is given to the detection cost. (Cdet(topic,bestVtx))$_{Norm}$ is the normalized detection cost for the topic as described in Section 5 and (Ctravel(topic,*bestVtx*))$_{Norm}$ is the normalized travel cost from the root vertex to the best vertex. Both values are normalized to be in the same "dynamic range" with 1.0 being the reference point. Otherwise, the linear combination via **WDET** would not be meaningful.

The variable 'bestVtx' is the vertex with the lowest combined travel and detection costs found by the search algorithm.

Ctravel() is the travel cost to the vertex from the root vertex.

Ctravel(topic,vertex) =

Ctravel(topic, parentOf(vertex)) +

CBRANCH * NumChildren(parentOf(vertex**)) +**

CTITLE

Where: Ctravel(topic,root) = 0

CBRANCH and **CTITLE** are evaluation constants. **CBRANCH** controls the bushiness of the DAG, and **CTITLE** corresponds to the cost of reading the cluster summary which controls the depth of the DAG. In combination, they have the effect of controlling both aspects of the DAG structure. The choice of values is arbitrary. However based on Allan et al.'s assertion that the braching factor of 3 is desirable, the values 2.0 and 1.0 for **CBRANCH** and **CTITLE** respectively prefer tertiary and quatriary trees.[13]

Since the Ctravel value will to a large degree be a function of the corpus size (the bigger the corpus, the larger the expected travel cost), the value needs to be normalized. To make Ctravel normalization comparable to Cdet normalization (see section 5.2), we should require a system to take advantage of information

[11] James Allan, Ao Feng, and Alvaro Bolivar, "Flexible Intrinsic Evaluation of Hierarchical Clustering for TDT," Proceedings of the Twelfth ACM International Conference on Information and Knowledge Management (CIKM 2003), November 2003.

[13] Based on the average travel cost to all leaf nodes in a minimal spanning, n-ary tree with 40000 leaf stories clusters.

Deleted: large

Deleted: normalize

Deleted:

Formatted: Font: Bold

Deleted: correpsonds

Deleted: [12].

Deleted: ()

Deleted: ,

Deleted: Based on Allan et al.'s observation that the optimal system should have a branching factor (**OPTBR**) of 3, Ctravel() is normalized by the expected travel cost to reach a leaf node of a minimal spanning tertiary tree.

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

in the stories in building the DAG in order to achieve a score below 1. For Cdet, this is done by normalizing scores relative to the prior probability of a story being on-topic. There is no clear analog of prior probability for Ctravel, making it difficult to determine a suitable normalization scheme a priori. The normalization scheme adopted for the 2004 dry run evaluation is based on one view of a "default" DAG, assuming that a hierarchical organization of topics is preferable to a flat, bushy structure.

The normalized travel cost $Ctravel_{norm}$ is computed as follows:

$Ctravel_{norm} =$

$Ctravel / (CBRANCH * MAXVTS * NSTORIES / AVESPT) + CTITLE$

Thus, the travel cost is divided by the average travel cost that would be obtained if a) all vertices are attached directly to the root (a one-level DAG); b) the number of stories per topic corresponds to the average observed in development data (AVESPT); and c) since a story may have multiple assignments, that the ratio of total vertices to total number of stories is the maximum (MAXVTS), assuming some reasonable ceiling on total graph size. The values for the 2004 evaluation are AVESPT = 88 (based on TDT4 multilingual test data) and MAXVTS = 3 (i.e. it is assumed that the system DAG will have no more than 3N vertices, N = number of stories in the corpus). NSTORIES is the total number of stories in the test set.

Parameter settings for the trial HTD evaluation are summarized in Table A1.

Table A1. **HTD evaluation cost parameters**

Parameter	Value
P_{target}	0.02
C_{Miss}	1.0
C_{FA}	0.1
WDET	0.66
CBRANCH	2.0
CTITLE	1.0
AVESPT	88
MAXVTS	3

To aid in understanding the behavior of this metric and of HTD systems, NIST will report the C_{Det} score separately for each system, as well as the Minimal Travel Cost. The C_{Det} score alone corresponds to the metric used for the Topic Detection task in previous years, although it will not be directly comparable because HTD allows clusters to overlap.

4.2 EVALUATION DATA

The evaluation corpus for HTD, as for other 2004 TDT tasks, will be TDT5.

TDT5 is being annotated using a time-limited procedure. Therefore, topics are not necessarily fully annotated. That is, there may be on-topic stories not labeled as such because of the time limits imposed on annotators. In principle, this may artificially increase false alarm rates but reduce miss rates. Since this the first TDT corpus with time-limited annotation, the empirical effects of this limitation on TDT metrics are not yet known.

The TDT schedule and resources do not allow for extensive annotation after systems generate their results on the evaluation corpus (as in the TREC paradigm). But there are resources for a limited amount of post-test annotation adjudication. This adjudication will emphasize items that may have been missed due to the time constraints on the initial annotation.

Deleted: (Ctravel(topic,vertex))Norm =

Formatted: Body Text, Indent: Left: 0", First line: 0"

Formatted: Font: Italic, Hungarian

Deleted: (topic,vertex) /¶

Formatted: Font: Not Bold, Italic, Hungarian

Formatted: Font: Italic, Hungarian

Deleted: **OPTBR** * $\log_{OPTBR}$(NumStories) +¶

Formatted: Font: Italic, Hungarian

Formatted: Font: Not Bold, Italic, Hungarian

Formatted: Font: Italic, Hungarian

Deleted: * $\log_{OPTBR}$(numStories))

Formatted: Font: Italic, Hungarian

Deleted: given

Deleted: ***OPTBR*** ... [1]

Deleted: To the extent feasible, NIST will explore alternative metrics. For contrast with the Minimal Travel Cost, NIST will focus on an alternative metric that is not sensitive to distance from the root. For example, the Expected Travel Cost metric suggested in Allan et al. computes the expected travel cost to find every story in a cluster, given any one story in the cluster. This algorithm will traverse the root vertex only if it is on the shortest path between two stories in a cluster. This metric is conceptually appealing because of its relationship to a search for all stories on a topic, given any one of the stories. A drawback is its computational complexity: finding the expected travel cost for a cluster requires computing the search cost over all members of the cluster, i.e., treating each member in turn as the given story. NIST will explore using a modified version of this metric that uses pruning or a non-linear cost function to limit search to a reasonable neighborhood of the given story.¶

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . . version 1.0,

APPENDIX B: IMPLEMENTATION DETAILS

1. SOURCE DATA

Sites that have not previously participated in TDT Evaluations should see the NIST TDT web site for instructions on how to register and arrange to get the TDT corpora from LDC. The URL is: http://www.nist.gov/speech/tests/tdt/tdt2004

The 2004 TDT Evaluation will use the TDT5 corpus as test data, in the format distributed by LDC. TDT5 test data will be in the same format as TDT4 test data, which is now available for use as development data.

2. EVALUATION I/O FORMATS

For continuing tasks, the source data formats and I/O formats are unchanged from previous TDT evaluations. These formats support newswire text processing using reference segmentations, as required for TDT2004. For the convenience of systems configured for previous TDT evaluations, NIST has retained support for audio file processing and automatic segmentation, even though these are not needed for TDT2004.

For the new task, Hierarchical Topic Detection (HTD), the output format uses XML. This change is motivated by the greater complexity of the Hierarchical Topic Detection output as well as the increasingly widespread use of XML.

The following types of input files are supplied for each evaluation task.

- A task specific file (or files) that specifies experiment conditions and gives the names and sequencing of the source files to be processed. The task specific index files are described in the following sections.
- A second index file that contains auxiliary side information that systems are permitted to take advantage of. The auxiliary information index file is an ASCII text file with one record per file. The format of this file is given in Table B1.

Table B1. **Auxiliary information index file**

Record Structure	Record 1-N: <Filename> <Source> <Language> <DateStamp>
Field Descriptions	**Filename**: The root source file name, without file type extensions or directory information. (e.g. 19980612_1931_2033_NYT_NYT)
	Source: Name of the data source. (e.g. ABC_WNT, CNN_HDL)
	Language: Language (ENGLISH, MANDARIN or ARABIC) of the original source file.
	DateStamp: Date and time when the data collection began for this source. The format is: "YYYYMMDD HH:MM:SS" (e.g. "19980612 12:30:14")

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

B.3 TOPIC TRACKING I/O FORMATS

Input data: Each tracking test will be directed by an experiment control file. This file will document the conditions of the experiment and will contain a list of index files for that experiment, one index file per topic. Topic tracking index files contain a list of topic training stories followed by a list of source files for which the target topic is to be tracked.

The **experiment control file** structure is as follows. First comes a header record that defines the experimental condition for which the control file is intended. The header record contains 4 fields:

Record Structure	Header Record: # <SourceType> <TrainLang> <TestLang> <N_tValues>
Field Descriptions	**SourceType**: The information source type. Always "nwt" for TDT2004.
	TrainLang: The language of the training stories: "eng" for English, "man" for Mandarin, or "arb" for Arabic
	TestLang: The language of the test source files: 'mul,nat' for multilingual texts, native transcriptions or 'mul.eng' for multilingual texts, English translations (for non-English languages)
	N_tValue: The value of N_t. For TDT2004, this field will contain one of the following four values: "1", "2", "4", or "V" (for variable).

Each subsequent data record in the experiment control file will contain the filename of a topic tracking index file, one filename per newline-separated record:

Record Structure	Record 2…: <Topic_tracking_index_file>
Field Descriptions	**Index_file**: The filename of a topic tracking index file.

The **topic tracking index** file structure is as follows. First comes a header record:

Record Structure	Header Record: # <Task> <PointerType> <Topic=N>
Des cri	**Task**: An indication of the TDT task to be run. For topic tracking the field will contain **TRACKING**.
	PointerType: The type of boundaries to be output by the system. For TDT2004, this value is not used. However, it must be present as a placeholder; it will always be **RECID.**[14]
	Topic=N: Declares the topic id number for which the test is to be run. (This is for documentation only. Allowable information for topic training is restricted to the indices of the training stories that follow.)

Then come $(N_t)_{Max}$ records containing the topic training stories, in chronological order. (There will be fewer than $(N_t)_{Max}$ records if there are fewer than $(N_t)_{Max}$ on-topic stories in the corpus.) Only the last N_t of these stories are to be used for training. When the N_t training condition is "variable N_t", all of the topic training stories are to be used. For this condition, special topic tracking index files will be provided that contain a variable number of training stories.

Record Structure	Record 2…($(N_t)_{Max}$ +1): # Topic_training_story <Story_ID> <Source_file> <Begin> <End>
Field Descriptions	**Story_ID**: A character string story identifier. (This is the TDT5 corpus "docno".)
	Source_file: The file name of the source data file containing the training story. For TDT2004, this is always an untagged text stream file.
	Begin: The word index (or time, if the source file contains sampled audio data) of the beginning of the story.
	End: The word index (or time, if the source file contains sampled audio data) of the end of the story.

[14] For English and Arabic text sources, the RECID's are equivalent to words. For manually transcribed Mandarin text sources, RECIDs are equivalent to characters. For ASR transcribed Mandarin sources, RECIDs are equivalent to words.

Formatted: Indent: Left: 0", Hanging: 0.38", Page break before

Formatted: No widow/orphan control, Don't keep with next

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

Then come the records that identify the source files to process. These records will have two fields:

Record Structure	Record $((N_t)_{Max} + N_n + 2)$…: <Source_file> <Begin>
Field Descriptions	**Source_file**: The file name of the source data file to be processed. For TDT2004, this is always an untagged text stream file.
	Begin: the word/character index (or time, if the source file contains sampled audio data) of the point in the source at which processing is to begin. [15]

Output data: The Topic Tracking task is to hypothesize points in the source stream where the target topic is discussed. Topic tracking systems will perform this task by outputting information about these hypothesized points to a file, one record for each putative discussion of the target topic. The first record in this file will contain five fields that specify information that applies globally to the whole file. These five fields will contain:

Record Structure	Header Record: <System> <Boundaries> <N_t> <Topic> <PointerType>
Field Descriptions	**System**: An alphanumeric character string that uniquely identifies the system being tested. (E.g., CDM_P05-8.v37)
	Boundaries: either **YES** or **NO**, where **YES** indicates that story boundaries are supplied to the system being tested and **NO** indicates that they are not. For TDT2004, the value should always be **YES**.
	N_t The number of stories used to train the system to the target topic.
	Topic: An index number in the range {1, 2, . . . ~100} which indicates the target topic being tracked.
	PointerType: The type of boundaries to be output by the system. The possible values are **RECID** or **DOCNO** for textual source data or **TIME** for source data in audio form.

Each subsequent data record in the file will identify the beginning point in the source stream of a judgment about whether the target topic is being discussed, along with an associated decision and confidence. This decision and confidence will apply to all subsequent source data until the point specified by the next output data record. These records will have four fields and will contain:

Record Structure	Record 2…: <Source_file> <Pointer> <Decision> <Score>
Field Descriptions	**Source_file**: The filename of the source file being processed. When the pointer type (via the header field designation) is **DOCNO**, this field is ignored, but a non-space data element must be supplied as a placeholder.
	Pointer: Indicates where in the source file the subject discussion commences.[16] For textual source data, Pointer is either the beginning **RECID** in the source file (in the range {1, 2, . . .}) or the story's **DOCNO**[17] attribute from the corpus. For source data in audio form, Pointer is the beginning time, in seconds.
	Decision: Either **YES** or **NO**, where **YES** indicates that the system believes that the source being processed does in fact discuss the target topic. **NO** indicates not.
	Score: A real number that indicates how confident the system is that the source being processed discusses the associated topic. More positive values indicate greater confidence.

[15] Processing begins at the very beginning of almost all source files. However, when the source file includes the end of the training data, tracking doesn't begin until after the last topic training story.

[16] Output records must be ordered so that each successive record's value of Pointer is greater than that of its predecessor, thus indicating the termination of the range of the predecessor record's Decision and Score.

[17] When **DOCNO** is used as pointer type, the decision and score apply only to the specified story. Therefore, **DOCNO** inventories must match the evaluation corpus's **DOCNO** inventory. Omitted **DOCNO** are assigned the decisions **NO** with a score of –9e99.

Formatted: Page break before

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

B.4 SUPERVISED ADAPTIVE TOPIC TRACKING I/O FORMATS

The I/O files and formats for supervised adaptive topic tracking are identical to those for topic tracking, except that there is one additional file: the reference file that contains relevance judgments, to be used for the supervised adaptation.

The reference file format is identical to that used in scoring tracking results.

The first line of the reference file contains a header record. The header record simply documents the source of the reference data, as follows:

Record Structure	Header Record: <TOPICSET annot_type=*S* version=*S* release_date=*S*>
Field Descriptions	**annot_type=*S***: *S* is an alphanumeric string that identifies the type of annotation.
	version=*S*: S is an alphanumeric string that identifies the version.
	release_date=*S*: S is an alphanumeric string that specifies the release date.

Each subsequent line in the file contains reference information for one document with respect to one topic. The format is as follows:

Record Structure	Record 2…: <ONTOPIC topicid=*S* level=*S* docno=*S* fileid=*S* comments="S">
Field Descriptions	**topicid=*S***: *S* is an alphanumeric string identifying the topic.
	level=*S*: *S* is either YES or NO. YES indicates that the story was judged relevant. NO indicates that the story was judged not relevant.
	docno=*S*: *S* is an alphanumeric string identifying the story.
	fileid=*S*: *S* is an alphanumeric string identifying the file containing the story.
	comments="*S*": "*S*" is an alphanumeric string in quotation marks, containing comments.

The reference file contains entries only for stories that were judged by a human annotator. If a given story was not judged with respect to a given topic, the reference file will contain no entry for that story with respect to that topic.

B.5 HIERARCHICAL TOPIC DETECTION I/O FORMATS

Input data: The topic detection test will be directed by an index file containing a list of source files for which topics are to be detected. Systems must process the source files in order of occurrence. The index files will follow this format:

Record Structure	Header Record: # <Task> <PointerType>
Field Descriptions	**Task**: An indication of the TDT task to be run. For hierarchical topic detection the field will contain **HIERARCHICAL_DETECTION.**
	PointerType: The type of boundaries to be output by the system. The only value used here is **DOCNO**.

Each subsequent data record in the file will identify a source file to process. These records will have only one field:

Record Structure	Record 2…: <Source_file>
Field Descriptions	**Source_file**: The filename of the source data file being processed.

Output data: The Hierarchical Topic Detection task is to automatically cluster a collection of stories into a directed acyclic graph (DAG). Hierarchical Topic Detection systems will construct a DAG over a collection of stories and record the DAG to an output file in XML format. The required XML format is specified in the following DTD:

```
<!ELEMENT htd ( vertexSet, edgeSet ) >
<!ATTLIST htd system NMTOKEN #REQUIRED>
<!ATTLIST htd rootVertex NMTOKEN #REQUIRED >

<!ELEMENT edgeSet ( edge* ) >

<!ELEMENT edge EMPTY >
<!ATTLIST edge destVertex NMTOKEN #REQUIRED >
<!ATTLIST edge srcVertex NMTOKEN #REQUIRED >

<!ELEMENT vertexSet ( vertex+ ) >

<!ELEMENT vertex ( story* ) >
<!ATTLIST vertex name NMTOKEN #REQUIRED >

<!ELEMENT story EMPTY >
<!ATTLIST story docID NMTOKEN #REQUIRED >
```

Figure B1 shows an example of a DAG of clusters represented in this format. The example in Figure B1 represents the DAG shown graphically in Figure A1 in Appendix A. Note that the names assigned to vertices are arbitrary and will not be used in scoring.

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

Figure B1. **Example Output DAG**

```
<htd system="CDM_P05-8.v37">
<htd rootVertex="a">
<vertexSet>
  <vertex name="a">
  </vertex>
  <vertex name="b">
    <story docID="s1"/>
    <story docID="s2"/>
  </vertex>
  <vertex name="c">
    <story docID="s3"/>
    <story docID="s4"/>
  </vertex>
  <vertex name="d">
    <story docID="s5"/>
    <story docID="s6"/>
  </vertex>
  <vertex name="e">
    <story docID="s7"/>
    <story docID="s8"/>
  </vertex>
  <vertex name="f">
  </vertex>
  <vertex name="g">
    <story docID="s9"/>
    <story docID="s10"/>
  </vertex>
  <vertex name="h">
    <story docID="s8"/>
    <story docID="s11"/>
    <story docID="s12"/>
  </vertex>
  <vertex name="i">
    <story docID="s13"/>
    <story docID="s14"/>
  </vertex>
<vertex name="j">
    <story docID="s15"/>
    <story docID="s16"/>
  </vertex>
</vertexSet>
<edgeSet>
  <edge srcVertex="a" destVertex="b"> </edge>
  <edge srcVertex="a" destVertex="c"> </edge>
  <edge srcVertex="b" destVertex="d"> </edge>
  <edge srcVertex="b" destVertex="e"> </edge>
  <edge srcVertex="b" destVertex="f"> </edge>
  <edge srcVertex="c" destVertex="f"> </edge>
  <edge srcVertex="c" destVertex="g"> </edge>
  <edge srcVertex="f" destVertex="h"> </edge>
  <edge srcVertex="f" destVertex="i"> </edge>
  <edge srcVertex="g" destVertex="i"> </edge>
  <edge srcVertex="g" destVertex="j"> </edge>
</edgeSet>
</htd>
```

B.6 NEW EVENT DETECTION I/O FORMATS

Input data: The new event detection test (called "first story detection" within NIST scoring software) will be directed by an index file containing a list of source files for which systems are to detect the first story of new topics. Systems must process the source files in order of occurrence. The file format is identical to the detection index file, with the only exception being the designation of the task. The index files will follow this format:

Record Structure	Header Record: # <Task> <PointerType>
Field Descriptions	**Task**: An indication of the TDT task to be run. For new event detection the field will contain **FIRST_STORY**.
	PointerType: The type of boundaries to be output by the system. The possible values are **RECID** for source data in text form or **TIME** for audio data.

Each subsequent data record in the file will identify a source file to process. These records will have only one field:

Record Structure	Record 2…: <Source_file>
Field Descriptions	**Source_file**: The filename of the source data file being processed.

Output data: The new event detection task is to determine whether or not each processed story is the first story of a new topic. The new event detection system will perform this task by outputting one record for each putative decision to a file. The first record in this file will contain four fields that specify information that applies globally to the whole file. These four fields will contain:

Record Structure	Header Record: <System> <Boundaries> <N_f> <PointerType>
Field Descriptions	**System**: An alphanumeric character string that uniquely identifies the system being tested. (E.g., CDM_P05-8.v37)
	Boundaries: Either **YES** or **NO**, where **YES** indicates that story boundaries are supplied to the system being tested and **NO** indicates that they are not. For TDT2004, this value should always be **YES**.
	N_f: The maximum deferral period allowed before a decision must be made.
	PointerType: The type of boundaries to be output by the system. The possible values are **RECID** for text stream segmentation or **TIME** for audio segmentation.

Each subsequent data record in the file will identify a putative decision, the point in the source stream of that decision, and a measure of the confidence in the decision. This decision and

Deleted: ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶

Deleted: 5

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

confidence will apply to all subsequent source data until the point specified by the next output data record. These records will have four fields and will contain:

Record Structure	Record 2…: <Source_file> <Pointer> <Decision> <Score>
Field Descriptions	**Source_file**: the filename of the source data file being processed.
	Pointer: Indicates where in the source file the new subject commences. For textual source data, Pointer is the index number of the specified word, in the concatenation of all story texts for the source file (in the range {1, 2, . . .}). For source data in audio form, Pointer is the specified time, in seconds.
	Decision: Either **YES** or **NO**, where **YES** indicates that the system believes that the source being processed does in fact discuss a new topic. **NO** indicates not.
	Score: A real number that indicates how confident the system is that the source being processed discusses a new topic. More positive values indicate greater confidence.

B.7 LINK DETECTION I/O FORMATS

Input data: The story link detection test will be directed by an index file containing a list of source files to process followed by a list of story pairs for which story link decisions are to be made. Systems must process each story pair independently, and in order of occurrence. The index files will follow this format:

Record Structure	Header Record: # <Task>
Field Descriptions	**Task**: An indication of the TDT task to be run. For link detection the field will contain **LINK_DETECTION.**

Next is the list of source files to process, in chronological order:

Record Structure	Record 2…N_{source}+1: # source_file <Source_file>
Field Descriptions	**Source_file**: The file name of the source data file to be processed.

Each subsequent data record in the file will identify a pair of story ids to process. The first story id will be the first (i.e., oldest) story, and the second story id will be the second (i.e., most recent) story. The list will be sorted by the first column, and then by the second column. Thus the second (and most recent) story in each subsequent record will be in monotonically increasing chronological order. These records will have only two fields:

Record Structure	Record N+3…: <Story_ID_1> <Story_ID_2>
Field Descriptions	**Story_ID_1**: The character string story identifier of the first story. This identifier is the source file and the TDT5 corpus "docno," separated by a colon.
	Story_ID_2: The character string story identifier of the second story. This identifier is in the same format as for the first story, namely the source file and the TDT5 corpus "docno," separated by a colon.

Output data: The story link detection task is to determine whether or not each story pair is on the same topic, or "linked." The story link detection system will perform this task by outputting one record for each story pair presented in the index file. The order of the decisions in the output file *must* match the index file exactly. The first record in this file will contain two fields that specify information that applies globally to the whole file. These two fields will contain:

Record Structure	Header Record: <System> <N_f>
Field Descriptions	**System**: an alphanumeric character string that uniquely identifies the system being tested. (E.g., CDM_P05-8.v37)
	N_f: The maximum deferral period allowed before a decision must be made.

Each subsequent data record in the file will identify a putative decision on a story pair, and a measure of the confidence in the decision. These records will have four fields and will contain:

Record Structure	Record 2…: <Story_ID_1> <Story_ID_2> <Decision> <Score>
Field Descriptions	**Story_ID_1**: The character string story identifier of the first story. (This is the TDT5 corpus "docno.")
	Story_ID_2: The character string story identifier of the second story. (This is the TDT5 corpus "docno.")
	Decision: Either **YES** or **NO**, where **YES** indicates that the system believes that the story pair is linked. **NO** indicates not.
	Score: A real number that indicates how confident the system is that the story pair is linked. More positive values indicate greater confidence.

Deleted: 6

Deleted: . HTD Task Definition and Evaluation Plan . page

Deleted: of

Deleted: . version 1.0,

OPTBR	3